

Framework para Desenvolvimento de Aplicativos

Application Express

Business Intelligence

Cloud Computing

Communications

Desempenho e Disponibilidade de Banco de dados

Data Warehousing

.NET

Linguagens de Programação Dinâmica

Embedded

Arquitetura Enterprise

Enterprise Management

Grid Computing

Identidade e Segurança

Java

Linux

Service-Oriented Architecture

SQL & PL/SQL

Virtualização

Sistemas

## Configurando SGA X PGA no Oracle Database

Por Fabio Prado

Postado em Setembro 2016

Revisado por Marcelo Pivovar - Solution Architect

Este artigo visa esclarecer um assunto que grande parte dos usuários me questionam no início dos treinamentos [Database Performance Tuning](#) (sem saber que iremos abordá-lo na última aula): **como configurar apropriadamente a SGA e PGA no Banco de Dados Oracle?**

Vou partir do princípio que você já sabe o que são as áreas de memória PGA e SGA no Oracle (ver Imagem 01). Caso não saiba, consulte a seção [Memory Architecture](#) no doc Database Concepts do site oficial da Oracle.

É importante ressaltar que essa tarefa apesar de não ser um "bicho de 7 cabeças", gera muita confusão, principalmente por causa da grande quantidade de parâmetros que existem nas últimas versões do Oracle para configurar a SGA e PGA (ver Imagem 01). Meu objetivo aqui neste artigo, é ajudar os DBAs (ou qualquer profissional que precise fazer este trabalho) a configurar essas áreas de memória com menos dificuldades e sem erros!

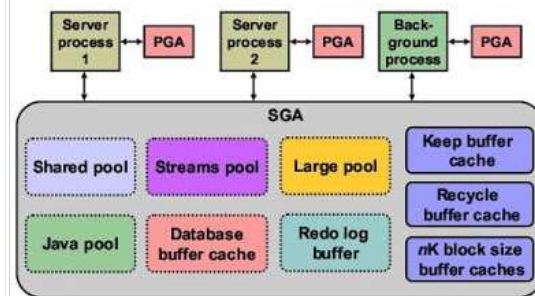


Imagem 01: SGA X PGA no Oracle Database.

Fonte: Oracle Corporation.

### Opções atuais de configuração da SGA x PGA:

#### 1- GERENCIAMENTO MANUAL (até o Oracle 9i)

Para efetuar a configuração do gerenciamento manual da SGA/PGA, o DBA tem que ter bons conhecimentos das subdivisões dessas áreas de memória e saber a quantidade de memória apropriada para cada uma delas, pois ele terá que configurar as principais delas, individualmente. Segue abaixo uma relação dos principais parâmetros que devem ser configurados no gerenciamento manual:

- **SGA:** DB\_BLOCK\_SIZE, SHARED\_POOL\_SIZE e JAVA\_POOL\_SIZE;

- **PGA:** SORT\_AREA\_SIZE, HASH\_AREA\_SIZE, BITMAP\_MERGE\_AREA\_SIZE e CREATE\_BITMAP\_AREA\_SIZE

Quem quiser aprender mais sobre estes parâmetros sugiro a leitura das referências ao final deste artigo, porém não recomendo a utilização dessa forma de configuração a partir do Oracle 10G. Apesar de ter começado a gerenciar Bancos de Dados Oracle na versão 10G, há uns 7 anos atrás configurei um dos Bancos de Dados de produção "menos críticos" que eu administro, com o gerenciamento manual para ver ele como ele funcionava. Resultado: me arrependi! Em pouco tempo começaram a ocorrer erros na instância por falta de memória na Shared Pool, o que me levou a configurá-la novamente com o gerenciamento automático de memória, que veremos a seguir.

#### 2- GERENCIAMENTO AUTOMÁTICO (a partir do 10G)

No Oracle 10G houve um grande avanço no gerenciamento de memória, que passou a permitir que o DBA configure apenas um parâmetro de dimensionamento para a SGA, e outro para a PGA, e as subdivisões delas passaram a ser gerenciadas automaticamente pelo próprio Oracle. Isso facilitou e muito a vida de um DBA, pois como comentei anteriormente, era muito comum, por exemplo, faltar memória em um determinado momento na Shared Pool, e isso gerar erros para os usuários do BD. O que o gerenciamento automático faz para resolver isso? Ele pode neste momento tirar memória da Buffer Cache e realocá-la para a Shared Pool. Isso é muito bom, pois a grande maioria dos servidores de Banco de Dados possuem memória escassa, menos do que o ideal, e essa realocação constante de memória, de uma subdivisão para outra, evita erros e utiliza de forma mais eficiente a RAM disponível para a instância do BD!

No 10G, a Oracle criou o gerenciamento automático da PGA e SGA (parâmetros SGA\_TARGET, SGA\_MAX\_SIZE e PGA\_AGGREGATE\_TARGET), e este último, ela chamou de ASMM (Automatic Shared Memory Management). Segue abaixo um script que deve ser utilizado como exemplo para configurar ASMM + gerenciamento automático da PGA:

```
-- Configurando ASMM + PGA automática (substituir xG pelo tamanho desejado)
alter system set sga_target = xG scope=spfile; -- tam. desejado da SGA
alter system set sga_max_size = xG scope=spfile; -- tam. máximo da SGA
alter system set pga_aggregate_target = xG scope=spfile; -- tam. desejado da PGA
```

No 11G, a Oracle criou um novo tipo de gerenciamento automático, que permite agora, configurar o tamanho da SGA e PGA através de um único parâmetro (MEMORY\_TARGET), e chamou isso de AMM (Automatic Memory Management). Segue abaixo um script que deve ser utilizado como exemplo para configurar AMM:

```
-- Configurando AMM (substituir xG pelo tamanho desejado)
alter system set memory_target = xG scope=spfile; -- tam. desejado
alter system set memory_max_target = xG scope=spfile; -- tam. máximo
```

E no 12c, nada mudou com relação ao gerenciamento de memória? Em quase todas as versões do Oracle, muda-se uma coisinha aqui ou ali, criam-se novas subdivisões de memória etc. No 11G, por exemplo, além do ASMM e o gerenciamento automático da PGA, foram criadas a [Database Smart Flash Cache](#) (11GR2) e a [Result Cache](#). No 12c, não ocorreram grandes mudanças significativas na arquitetura de memória do Oracle (exceto pelo fato da SGA e PGA serem compartilhadas por múltiplos Bancos de Dados na nova arquitetura Multitenant), mas foram criados, por exemplo, o [Automatic Big Table Caching](#), o [Force Full Database Caching Mode](#) e o parâmetro [PGA\\_AGGREGATE\\_LIMIT](#). Este último, criado para delimitar o tamanho máximo da PGA automática, recurso que antes não existia e cuja ausência gerava problemas em alguns ambientes de Bancos de Dados.

#### 3- CONSIDERAÇÕES IMPORTANTES

- 1- Na versão 10G, ASMM + PGA automática são o método de configuração padrão de memória sugeridos ao utilizar o DBCA;
- 2- Na versão 11G, se RAM menor ou igual 4G, o DBCA sugere AMM, caso contrário, ele irá sugerir ASMM + PGA automática;
- 3- A partir do 11G, ao configurar ASMM + PGA, deve-se zerar os parâmetros relativos ao AMM, como no exemplo do script abaixo:

```
alter system set memory_max_target = 0 scope=spfile;
alter system set memory_target = 0 scope=spfile;
```

4- A partir do 11g, ao configurar AMM, deve-se zerar os parâmetros relativos ao ASMM + PGA automática, como no exemplo do script abaixo:

```
alter system set sga_max_size = 0 scope=spfile;
alter system set sga_target = 0 scope=spfile;
alter system set pga_aggregate_target = 0 scope=spfile;
```

**CUIDADO:** Se os parâmetros acima não forem zerados, ao configurar AMM, eles serão utilizados como valores mínimos e máximos para a SGA e PGA. Inclui-se isso é tema de perguntas da certificação [Oracle Database 11g Performance Tuning Certified Expert](#). Outro ponto interessante na configuração de AMM, é que em ambientes de 64 bits, o valor de `sga_max_size` é ajustado automaticamente para ficar igual ao maior valor dos parâmetros `memory`.

5- Os parâmetros `sga_max_size` e `memory_max_target` não são dinâmicos, portanto, se for necessário alterá-los você precisará reiniciar a instância.

Muitos podem agora estar se questionando: qual o tamanho ideal dessas áreas de memória? A resposta é longa, mas é muito importante, pois SGA e PGA subdimensionadas, ou até mesmo superdimensionadas, não são boas para o desempenho do Banco de Dados. Não entrarei neste detalhes neste artigo, porém sugiro que você comece analisando os itens abaixo:

1- Verifique os advisors de memória no AWR ou consulte diretamente as visões de performance dinâmicas relacionadas, como no exemplo abaixo:

```
-- verificar se aumentando memória teria melhora de desempenho (estd_db_time)
Select * from v$memory_target_advice
order by memory_size;
```

2- Verifique se estão ocorrendo muitas operações de redimensionamento de memória automáticas. Se sim, considere aumentar a memória da SGA:

```
select component, oper_type, oper_mode, parameter, initial_size, final_size,
to_char(start_time,'dd/mm/yyyy hh24:mi:ss') start_time, to_char(end_time,'dd/mm/yyyy hh24:mi:ss') end_time
from v$memory_resize_ops;
```

3- Para tunar a PGA, na consulta abaixo verifique se "ESTD\_OVERALLOC\_COUNT" > 0. Se sim, considere aumentá-la:

```
SELECT round(PGA_TARGET_FOR_ESTIMATE/1024/1024) target_mb,
ESTD_PGA_CACHE_HIT_PERCENTAGE cache_hit_perc,
ESTD_OVERALLOC_COUNT
FROM V$PGA_TARGET_ADVICE;
```

Por hoje é só! Espero que você tenha aprendido algo e que o artigo lhe seja útil!

#### Referências:

[Database Administrator Guide, Management Memory:](#)  
<https://docs.oracle.com/database/121/ADMIN/memory.htm>

**Fabio Prado** atua na área de tecnologia desde o ano de 2000, possui experiência e profundos conhecimentos em Análise de Sistemas, programação (Dot Net) e Administração de Bancos de Dados Oracle. É DBA em uma empresa do governo, instrutor na empresa de treinamentos FABIOPRADO.NET, Oracle ACE Member, Organizador do DBA BRASIL e autor do blog [fabioprado.net](#).

*Este artigo foi revisto pela equipe de produtos Oracle e está em conformidade com as normas e práticas para o uso de produtos Oracle.*

 E-mail this page  Printer View

## Entre em Contato Conosco

Vendas: 0800-891-4433

Contatos Globais

Diretório de Suporte

## Sobre a Oracle

Informações sobre a Empresa

Comunidades

Carreiras

## Nuvem

Visão geral de Soluções em  
Nuvem

Software (SaaS)

Plataforma (PaaS)

Infraestrutura (IaaS)

Dados (DaaS)

Teste em Nuvem Grátis

## Eventos

Oracle OpenWorld

Oracle Code

JavaOne

Todos os Eventos da Oracle

## Ações Principais

Faça o Download do Java

Faça o Download do Java para  
Desenvolvedores

Experimente o Oracle Cloud

Inscreva-se para Receber E-mails

## Notícias

Redação

Revistas

Histórias de Sucesso de Clientes

Blogs

## Tópicos Principais

ERP, EPM (Finanças)

HCM (RH, Talento)

Marketing

CX (Vendas, Serviço, Comércio)

Soluções Setoriais

Banco de Dados

MySQL

Middleware

Java

Sistemas Projetados